

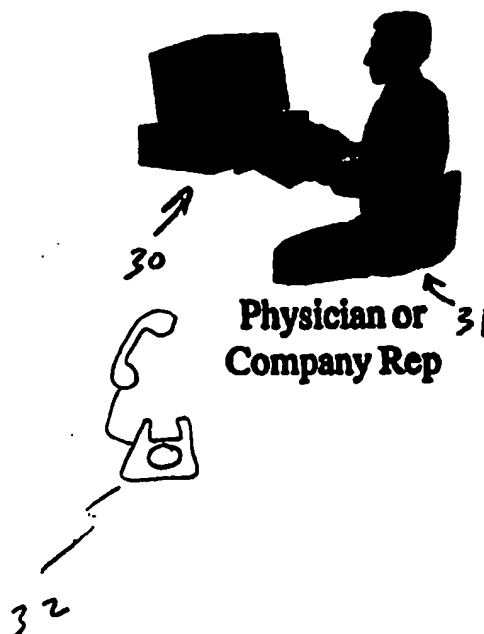
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : A61N 1/37	A1	(11) International Publication Number: WO 98/42407 (43) International Publication Date: 1 October 1998 (01.10.98)
(21) International Application Number: PCT/US98/06085 (22) International Filing Date: 27 March 1998 (27.03.98) (30) Priority Data: 60/042,367 27 March 1997 (27.03.97) US (71) Applicant (for all designated States except US): MEDTRONIC, INC. [US/US]; 7000 Central Avenue Northeast, Minneapolis, MN 55432 (US). (72) Inventors; and (75) Inventors/Applicants (for US only): WEBB, James, D. [US/US]; 8138 Ranier Lane, Maple Grove, MN 55311 (US). NELSON, Chester, G. [US/US]; 6755 Cottonwood Lane, Maple Grove, MN 55369 (US). THEIS, James, G. [US/US]; 11015 Louisana Lane North, Champlin, MN 55316 (US). STAUFFER, Ronald, A. [US/US]; 10784 Kumquat Street Northwest, Coon Rapids, MN 55448 (US). (74) Agents: ATLASS, Michael, B. et al.; Medtronic, Inc., 7000 Central Avenue Northeast, MS301, Minneapolis, MN 55432 (US).		(81) Designated States: CA, JP, US, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

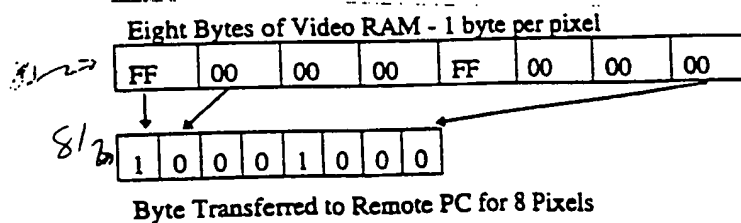
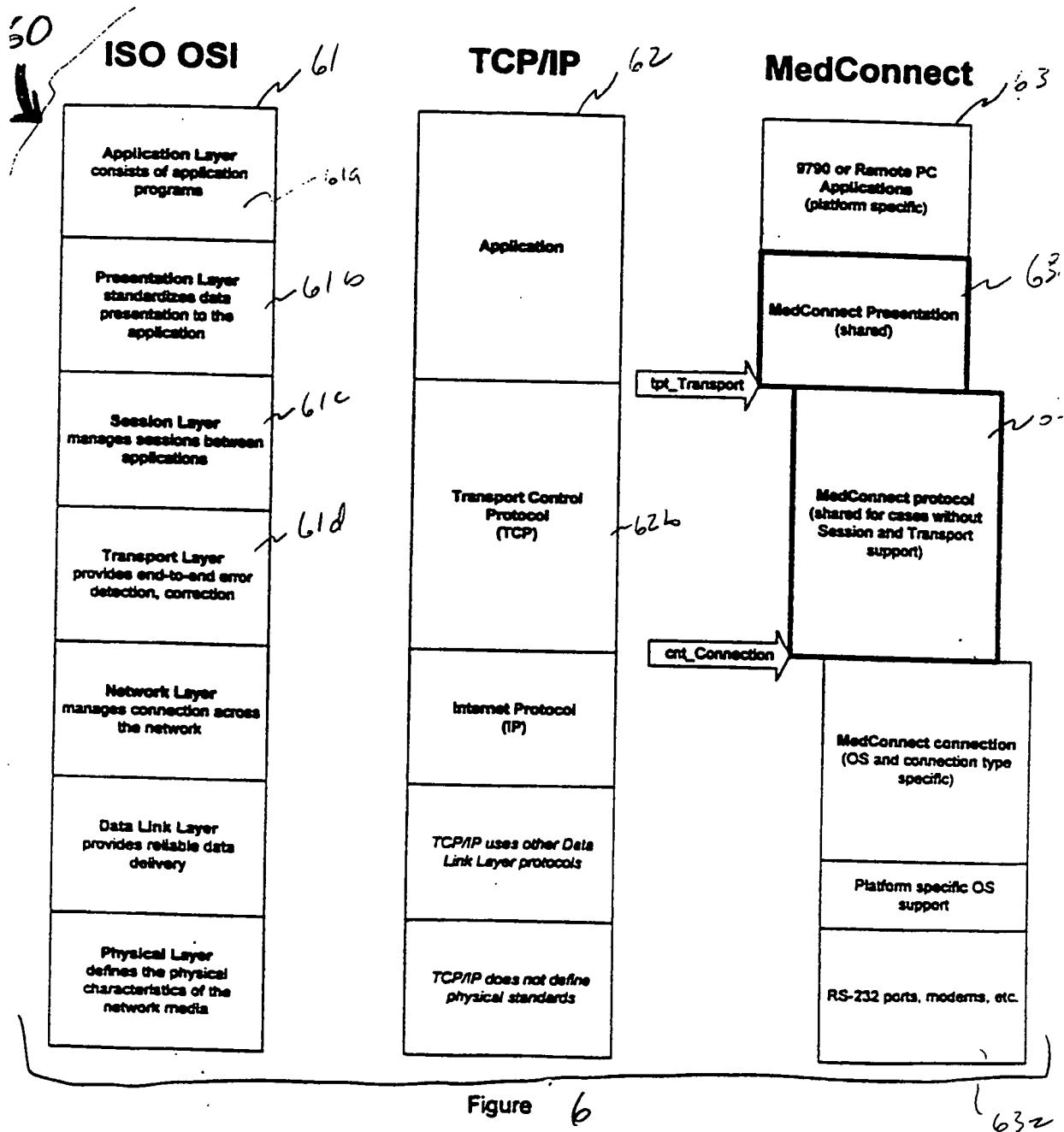
(54) Title: IMPLANTABLE MEDICAL DEVICE REMOTE EXPERT COMMUNICATIONS SYSTEM FOR COORDINATED IMPLANT AND FOLLOW-UP

(57) Abstract

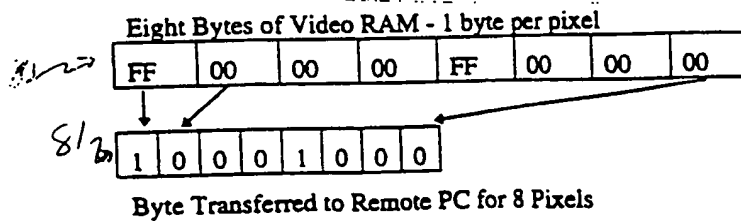
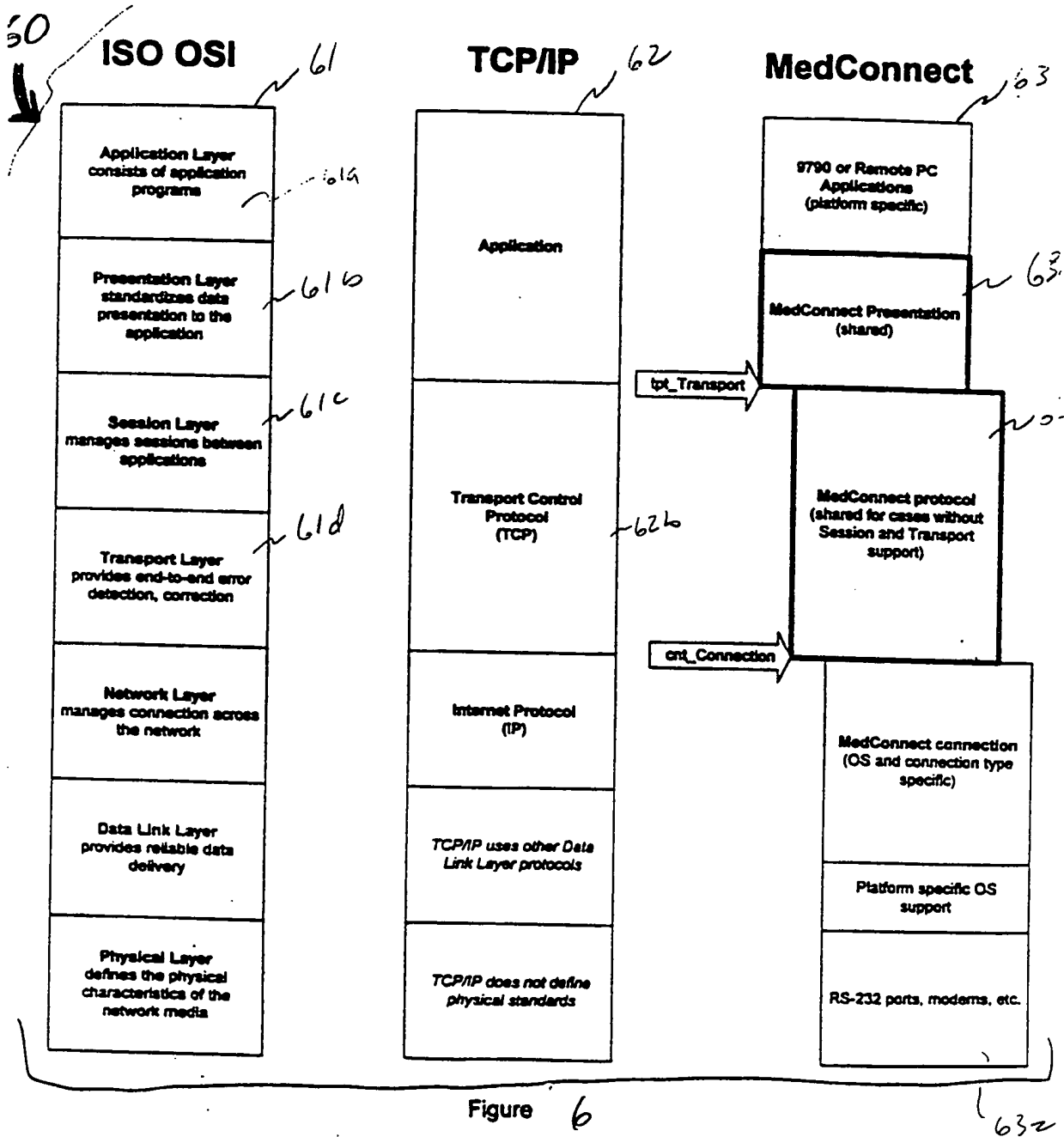
A medical device implant, monitoring and adjustment are enhanced by the telepresence of a remote expert (31) having a screen display (30) that mirrors the display at the patient location. Pointers (51), whether they can activate subprograms at the patient location or not, are moved identically at both locations at the same time. Also, the ECG and any features thereof are updated in real time, while other communication operations occur.

Remote "Experts"

5/11



5/11



	Epard et al,	US Pat. No	5,241,625,
	Inga et al.,	US Pat. No	5,384,643,
	Thompson	US Pat. No	4,377,852,
5	Ziv-El	US Pat. No	5,263,869,
	Thessin et al.	US Pat. No	5,452,299,
	Harlan	US Pat. No	5,168,269,
	Torok, et al.	US Pat. No	4,317,956,
	Stein et al.	US Pat. No	4,098,267, and
10	Bijnagte	US Pat. No	5,235,680.

There have been remote device operations used before, wherein a remote expert could control what was occurring on a local technicians station, including for example, the model 89441a Vector Signal Analyzer(TM) from Hewlett Packard, but no prior devices of which the inventors are aware include the features of this disclosure which are particularly suited to remote expert supervision of a local implantable medical device operation. The instant disclosure relates particularly to allowing for repeatable remote near real time ECG, EGM, and marker channel information at the same time display and allowing for dual site manipulation of a pointer on the display at both locations without having conflict. [In this document we will use various terms such as ECG, EGM, EKG and Marker Channel information to represent that such types of information as the intracardiac electrogram or surface ECG's, with or without additional marker channel information are all signals that could be found in the display window from whatever physiologic signal that the local attendant and remote expert may desire to look at simultaneously. Accordingly, any such use of these terms should be assumed to apply to the use of any of these terms for any similar physiologic signals, with or without device information superimposed thereon, such as Marker Channel information.]

The primary use of the invention is believed to be pacemaker/defibrillator follow-up and implant where an expert is remote from the patient's local situs where such operations are occurring. What is believed to be needed is a communications system for communicating in real time or near real time the position of a pointer that is manipulable at more than one location on simultaneous displays at two locations so that the pointer will appear at the same location in the remote display as the local display regardless of any conflicting instructions given by users at the remote and local locations. Provision is made to allow for simultaneous activation of two pointers, one controlled by the expert and one by the attending technician/physician. Each has a unique icon in our preferred system embodiment.

This system should have simultaneous displays for communicating and where necessary, enhancing the ECG signal available for remote display such that it appears to move like a normal ECG at the remote location with minimal difficulty and movement occurs in near real time. Of particular concern in using typical remote viewing systems is that screen displays occasionally have lossy displays, so a short term change in a physiologic signal may be missed at the remote location. The sweep-bar or smooth movement of the signal should ideally be maintained. The availability of such a system would help a great deal in surgery and particularly cardiac surgery. It could also find application to the cardiac pacemaker clinic follow up. With a coordinated cursor or pointer only one of the remote and local actors can adjust its position at a given moment, and there can be no confusion as to what location in the display they are both communicating about through some other telephonic aural or video link which should be operative simultaneously to best employ the inventive features described herein.

What such a system would enable is the review by an expert on either the patient (as in a doctor, for example) without this expert needing to be present in real space. It alternatively and simultaneously could be used for the consultation of a local physician or surgeon with a systems expert that may have intimate knowledge

about the components of the devices that are generating the EKG signal and performing other functions in the surgical or clinical locality.

Brief description of the drawings

5

Fig. 1 is an overview of the " local" environment having a physician, patient, and programmer device present.

Fig. 2 is a simplified representation of any data communications system.

10

Fig. 3 is an illustration of a distant expert and his computing device.

Fig. 4 uses a communications layer block diagram illustrating " MedConnect " layers and related communications layers in accord with a preferred embodiment of the invention.

15

Fig. 5 is an alternative view of the communications interface illustrated in Fig. 4, expanding on the details of the communications layer depicted in Fig. 4..

20

Fig. 6 is a stack diagram of communications and protocol layers providing an alternative representation of Fig. 5.

Fig. 7 is an illustration of a two color (8 bytes to 1 byte) video data compression scheme used by a preferred embodiment.

25

Fig. 8 is an alternative graphic representative illustration of the relationships of the communications linking arrangements displayed in Fig. 6.

Fig. 9a illustrates an ECG display window which would be a single element on a display screen in accord with a preferred embodiment.

30

Fig. 9b. illustrates three data packets that may be organized as shown for preferred embodiments of the invention.

5 Figs. 10a and 10b are block diagrams illustrating how program controls provide coordination between remote and local ECG data.

Figs. 11-14 are illustrations of screen displays in accord with preferred forms of this inventions.

10 Fig. 15 is a heuristic diagram of system components and their interaction in accord with a preferred embodiment.

15 Fig. 16 is a heuristic diagram having a blown up portion illustrating the arrangement of video RAM data memory organization relative to an area of video display.

Summary of the invention

20 Various aspects related to remote communications are herein described with particularly here and claimed, those improvements provide a new system image the features of a moving remote ECG or other physiologic signal display and real time remote pointer coordination.

25 In general a personal computer with a microprocessor and memory that contain appropriate software to configure them to operate as detailed herein is connected to a programmer which has similar and interrelated components, via any one of a number of methods including modem connections over telephone lines, DSVD modems, wireless, or internet, or any other similar form of communications
30 link that may be in use and easily available. In other words, the system would

employ a set of at least two devices having a display at a remote and at an operating location and communications pathways between, that communicate a coordinated display of information on the remote location display and in the operating location display such that communications are enhanced between the users at the two locations by use of the two coordinated display means at the remote and local operating (i.e. programmer) locations. The local display may also display local information to the local user that is uncoordinated with or completely separate from information displayed at the distant site display if desired. For examples, a battery power available display may appear on a local portable device display, that is unnecessary for the distant location to be aware of; or an expert system window could be available to the local display of the remote expert actor which is unavailable to the display at the location of the operating room surgeon.

At one end is the programmer device for use in medical communications systems for communicating in near real time information from between at least two remote sites, one remote site at a patient location having the programmer device which has a programmer generated display, and an other of said at least two remote sites at an expert location having a computing device with a computing device generated display, each location having available substantially similar screen displays, such that information related to an implantable medical device and a patient at said patient location remote site can be reviewed simultaneously on said substantially similar displays at said at least two remote sites facilitated by data communications transferred across a communication line between said at least two remote sites, said programmer device being configurable to perform various operations by executing programs through a microprocessor and memory system although many of the programs could be converted to hardware having limited functionality. This programmer then would have a physiologic signal device connection, connected to said programmer device for receiving electrical output representing a patient's physiologic signal waveform ,and for producing a signal suitable for use by a programmer screen display operating program for generating a

display from said electrical output at at least one display area including at least a physiologic waveform signal display area and producing instantaneous changes in said at least one display area, at least one of said changes representative of changes in said waveform. The programmer display operating program, would use a number of elements including at least

a first video display data buffer for holding a video data representation of information displayed in one of said at least one display areas of said programmer generated display,

a second video display buffer for holding data representing instantaneous changes in at least one of said areas,

a display generating program for using a data image of said video data representation in said first video display buffer for producing said programmer generated display, and it would also have

an index generating program for reviewing data in said first video display buffer and creating a first index buffer corresponding to said at least one of said display areas, said index generating program indicating in said first index buffer by storing data therein representing which of all locations in said at least one area have changed from a previous review, and for capturing a data representation of said changed locations in a capture buffer and, a sending program for packaging data comprising contents of said first index buffer and said capture buffer in appropriately encoded format, and for sending said appropriately formatted index buffer data across the communication line so as to get the information to the other remote site(s).

Preferably, the display generating program also has a two color display generating program for interpreting said physiologic signal waveform as two color display data, and wherein said index generating program has a data compression program for receiving said physiologic waveform index data and coding it such that for each byte, one bit is set to indicate whether a change has occurred in that byte

and such that for the entire physiologic waveform index buffer, a compressed physiologic waveform index is prepared.

The programmer operating program preferably also has a pointer/cursor coordination program having a buffer for holding at least location information for each pointer/cursor to be coordinated, and having a monitoring means for updating any programmer generated display change in pointer/cursor activation and location, said monitoring means updating each said pointer/cursor buffer with any change in location information from a last review, said location information being changeable responsive to pointer/cursor user controls based on a local user changing location status of any pointer/cursor under local control, and wherein said pointer/cursor coordinating program further comprises a buffer for containing information regarding at least location status of each remote pointer that is to be displayed on said patient location display. The programmer operating program having these pointer/cursor features operates with the patient location screen display operating program, which to work with these pointer/cursor features has a program for receiving new information for each said remote pointer/cursor buffer and operating on such data to produce a display of said remote pointer/cursor in accord with the data in said remote pointer/cursor buffer such that any changes in the location data of a remote pointer/cursor buffer results in a change in location on the display of that remote pointer/cursor on the patient location display screen.

Preferrably, the pointer cursor operating program also has an activation buffer for maintaining activation status for each pointer/cursor capable of activation. But if activation is not available from a remote pointer/cursor, such a buffer would not be needed for the activation status of that particular icon, although it could be useful information and used at the programmer site to inform the programmer user that the expert has indicated that the expert thinks this particular pointer/cursor should activate a particular function, even if it is not activatable by the expert

communicating from the remote site. In some embodiments the expert will be able to remotely activate features on the programmer though such a feature.

Of course, to activate the features of pointer/cursors we prefer to include in the programmer device a pointer data capture program for capturing and then for sending the data in said pointer/cursor buffer to said communication line.

Additionally, we prefer to include a program for monitoring which of all said pointer/cursors is active such that said active cursor/pointer may initiate a command, and for maintaining a data buffer containing such monitored information.

Preferably, the computing device operating program for coordinating a physiologic waveform display with a programmer generated display will have a communication means to receive data sent by said programmer device and to produce a physiologic waveform display therefrom.

On the other side of the system(s) we prefer a computing device operating program for displaying an image substantially the same as the display on the programmer generated display having communications means to receive data sent by the programmer and generating a substantially similar display therefrom.

This of course preferably comprises a system for displaying substantially the same image on at least two remote locations having a programmer device and a computing device as set forth above.

The invention, then, can be described as a method for communicating between at least two remote locations so as to enable coordinated display of a physiologic waveform and pointer/cursor information which preferably has these steps:

capturing pointer/cursor information regarding a location on a display screen for each said pointer/cursor,

redacting image data from a video display buffer based on said pointer/cursor information so that the remaining video buffer data represents a display sans said pointer/cursors

reviewing the redacted video image buffer so as to locate any changes since a last review,

creating an index of information describing which locations in said video buffer contain changed data since said last review,

capturing representations of the images at the changed locations and storing this as data in a buffer,

compressing the index and the capture buffer data,

sending the compressed index and capture buffer data from one of said at least two locations.

Preferably, said pointer/cursor data is sent in said sending step.

It can also be described as a method of communicating between at least two remote locations so as to enable coordinated display of a physiologic waveform and pointer/cursor information preferably having these steps:

receiving compressed index and capture data across a communication line from one of said at least two locations at a first of said at least two locations,

decompressing said index and capture buffer data,

swapping in information from said capture buffer data at locations in a video display buffer representing corresponding locations to those indicated in said index data so as to enable a display at said first location to display said changes.

A preferred form of the method includes in the step of receiving data representing pointer/cursor location on a display screen from said one location of at least two remote sites and a further step of displaying an iconic image of ca

pointer/cursor representing a pointer displayed at said one remote location in a location on the screen in said first locations indicated by the received representative pointer /cursor location data.

Detailed description of preferred embodiments

It should be noted that several of the terms used within this document are peculiar to particular companies systems. For example, the term 9790 is the name of a programmer used for programming implantable devices produced by Medtronic, Inc and well known in this field. The 'Macintosh' or 'Mac' is the name of the operating system/user interface for Apple Computer company's computers. E C G and E K G are abbreviations discussed in the background section. Minix 8340 is a name for a particular implantable pacemaker produced by Medtronic, Inc.

Numerous terms are also employed which have common computer and communications industries meanings including, PC, meaning a small or "personal" computer, " application", "layer", TCP, RS-232, and other terms have their commonly used and known meanings also.

In its broadest form, the invention operates in a system having at least those parts heuristically illustrated in Figs. 1-3. In fig. 1, a programmer 10 (a Medtronic 9790) operates as the local device having a display 10a and input mechanisms for use by a clinician 12 who may be examining a patient 11. fig. 2 briefly illustrates that communication channels may include phone lines 20 to provide data transmission as well as voice communication between the device illustrated in fig. 1 and the device of fig. 3, a physician, company representative, or other expert 31 one may be used the communications device 30. In both Fig. 1 and in Fig. 3, are device 15 and 32, respectively, which illustrate that an additional communications device and/or channel is useful for the best operation of inventive system.

Fig. 4 illustrates in diagram form, a version of the same system 40 as was described with respect to Figs. 1-3, as essentially of two parts, a part 41 local to the expert but remote to the clinician, and a part 42 local to the clinician but remote from the expert. In the preferred embodiment, connects between the pacing expert 31 and the remote user 12 via modem 15, 17 or any other type of modem. Here, a cellular phone 16 may connect to the modem instead of using wire connections as just another example of the realm of possible system communications components. Video and/or audio connections 13,14 may additionally be maintained to facilitate the communications between the user and the expert. With this setup, the data on the remote access system 21 at the expert's site can send and receive data, be in communication with the "VTRX" and other programmer software applications running at the remote user site. (VTRX is a particular kind of system like a DOS or Windows or OS/2 operating system from Microsoft or IBM, but it has some features for running applications in programmers not normally found in the other OS's). There does need to be some coordinating program, here called remote access 18 at the user site that is designed to understand the suitably configured data streams sent between the remote access program at the expert site and the data streams and programs that use them at the user site.

It is important to note that there are various levels of communication required in order for the user accessible high level display programs in the remote access device 21 and the related programs in the expert device to be coordinated. In Fig. 5, the system 50 is illustrated having a logical facilitated communications link 51 between each of the operating layers of the displays at each location. For device 41a and device 42a, a software layer management connection 54a and 54b respectively, must be communicating in a cooperative manner. Additionally, a communications control layer 53a and 53b must pass the working information from the connection management layer to the capability layer 52a and 52b, in order for the cooperation to be accomplished.

A more typical figure for explaining the interrelationships between protocol layers is in Fig. 6. Referring now to Fig. 6 it first must be recognized that the entire column for the preferred MedConnect system would exist at both the expert and patient sites. Remote access software at both sites is depicted in block 63. The columns to the left of the "MedConnect" column depict mappings to other existing communications architectures that may alternatively be used or which may communicate with the MedConnect system. Blocks 63a and 63 together equate to the TCP/IP concept of an Application block 62 or the ISO/OSI concepts of Application layer 61 and Presentation layer 61b. Block 63b equates to the TCP/IP concept of the Transport Control Protocol or the ISO/OSI Session Layer 61c and the Transport Layer 61d.

Also, as would be understood by communications workers familiar with such diagrams, the column under "MedConnect" in most embodiments names the layers of hardware and software from the user application to wires and back through which any data must flow. In other words, the application layer passes data through each layer beneath it and from one set of wires at layer 63z, back up into another identical column at the remote location to block 63 in the most preferred or MedConnect to Medconnect embodiment. It is also recognized that not all uses will be of the proprietary MedConnect protocol so the communication to applications in columns 62 and 61 can also occur, although through different layers as shown in those columns. Thus in one preferred form, it can be seen at 61 that a feature in the application layer (block 61a) of the expert system located remotely from the application layer in (the 9790 or similar patient local device) in column 63 must pass data on timing and location that survives numerous translations and communications down into the physical layer across the communications network and back through another physical layer to the remote application at the top of its column. Of particular relevance to the instant invention are blocks 63a, and 63b, which enable the coding and decoding of appropriate information to accomplish the specific inventive coordination tasks. Thus, the presentation layer 61b in the ISO

OSI standard system on the right would contain the decoding and encoding software features necessary to accommodate layer 63a data, and if it did employ session data, it would have to coordinate with the connection layer 63b data in layer 61c and coordinate transport data through layer 61d. The lower layers would just carry the packets and physical signals and should, of course be coordinated in well known manners. However, in the TCP/IP case, the layer 63b could be eliminated as unnecessary. Also, if both systems had all the elements of the MedConnect system column, they would both need the layer 63b. To summarize, there must be a similar protocol stack in each location's system.

Referring now to Fig. 8, this diagram provides an alternative view of one organization for the preferred embodiment communication layers from 86 through 89 at a location within one of our preferred embodiment systems. At the upper level, the application that provides the device functionality at the device site, say for example, enabling pacing system analyzer data to be monitored and used at an implant where leads are connected for testing to a programmer. The application software would control the display on the local screen within the application layer by using the data in the presentation layer 87. In this layer, the display or "prt" cloud 84 contains data used for location and other display functions, (color etc.,) and this data is communicated between the application and the cloud 85(ex by direct links 81 to these particular data values (prt_ReportMsg, prt_PointerMsg, and prt_ScreenDataMsg, among others which are not shown). Each piece or chunk of data for communication can be thought of as being placed into a buffer for formatting appropriately for enabling the communication of that data in the buffer. These buffers are combined into the data blocks or packets shown for example in Fig. 9b. These values are linked to the transport and connection layers as shown. In understanding this diagram of Fig. 8, recognize that the arrows indicate that the cloud doing the pointing is a subset of types that are represented by the general cloud type pointed to.

Application threads 85 exist on both the expert and patient local devices.
This layer 86 is equivalent to block 63 in Fig. 6.

The applications 85 use the values in these components (e.g.
prt_ScreenDataMsg, prt_PointerMsg, prt_ReportMsg) to format data
communications messages between the Expert and Patient devices.
These components are of a base type prt_MedConnectMsg. All these message types
would be contained in the MedConnect Presentation block 63a depicted in Fig. 6

Details below the Transport layer line 88 relate strictly to communicating the
data between the expert and patient device locations.

The tpt_Transport base type equates to the block 63b depicted in Fig. 6

The cnt_Connection base type equates to the blocks below cnt_Connection
arrow on Fig. 6.

For specialized different types of communications media there are
representations given (e.g. cnt_ clouds pointing to base type cnt_Connection
cloud 84c) each of which uses generalized abilities of the Base type.

In Fig. 9a, an ECG display window 90 is illustrated, the height of the
window 93 is constant throughout, and the window itself comprises essentially two
segments, an active portion 92 and a trailing portion 91. Only a small part of
window 90 is actively receiving new data at any time in either the remote or local
device. It is the nature of ECG wave form that only the leading edge is active, the
rest of the display merely allows the leading edge to travel rightward along the
screen. Therefore only the active portion needs to be maintained through
coordination, since the display manufacturing software local to either system can
simply redraw the element that was in 90 as it moves to the right in any of

numerous well known ways. Thus, while the local device is receiving the actual electrocardiogram from the patient's attached electrodes, it is sending this data to the remote device as it also draws it on the local screen. The amount of data to transfer is very small since only the leading edge needs to be sent. The refresh rate will depend on how wide window 92 is in the preferred embodiment, the baud rate for data transfer(, the size and frequency of the data packets being transferred will affect the baud rate), the speed of the local computing and display device, and the resolution of the image.

However, given sufficient bandwidth, the entire ECG (or other physiologic data signal) display can be transmitted continuously.

Fig. 7 details a data transfer configuration employed by a preferred embodiment, and an explanation of how it works will help to indicate how the data of a screen display is transferred. When used for the ECG display area, we found that the simplest way to transmit a lossless physiological signal was to limit the display to a two color display, such as Black and White. In doing so we could transmit the entire changed area of the screen representative of the physiologic signal that exists at the moment by only transmitting those very small portions that change. Thus the entire line of ECG data could be transmitted in a very short chunk of data. The video RAM (Random Access Memory, a term with its common industry connotations), is divided into indexed area representations each of which indicate whether and what pixel configuration is lit, colored, and so forth. These typically require a full byte for an 8 color pixel and significantly more for any more enhanced pixel representations(like 16-color displays for example). By limiting ourselves to two colors, the pixel is either lit or not lit, thus inherently allowing for a quick compression to a one or zero representing the pixel. Also, by using run length encoding, every byte of identical data becomes a single bit in the transmitted data stream. So referring back to Fig. 7, we have divided the video RAM index into 8 bytes for the area under consideration. If there is a change in any of these

areas, a corresponding bit is set in the data word 81 which will be sent to the remote location indicating which of the areas of the index have changes in the display at the location known to be indicated by that index. Accordingly, the program at the receiving end will expect, for this example, the first and fifth sections need to be updated at its location and will look to the data in the remainder of the data packet or packets being transferred which relate to this index byte 81 for details about changes in those indexed locations. Typically, the X,Y location of each byte will be kept in a look-up table in both locations so the machines at each location can interpret this data as it comes in. The area of the screen which contains the display of the non new portion of the ECG will not be indexed, since it will be computed as the follow through of the leading edge information always received in each information transfer that has a new index byte, in the preferred embodiment.

In general, continuous screen capture is done by repetitively examining video memory, comparing it to the last known state of video memory, and then sending only the data representing the rectangular regions where data has actually changed on the screen. The video memory is read directly and the software screen compare routine finds the changed rectangles. We employ a specific algorithmic process to accomplish the search for changed rectangles set forth in detail in the Appendix, hereby incorporated by reference. The result of using this preferred program (although any one which accomplishes the result would suffice) is illustrated with reference to Fig. 16, wherein a portion of a videodisplay representing the ECG signal display 160 is shown having a one color physiologic waveform line 161 and alternate color surrounding space 162 drawn thereon. (The use of this program or a similar one could also be applied to the rest of the screen but our most pressing concern is for rapid updating of the physiologic waveform data so this is done separately, preferably). The rectangle area 163 is represented by an 8 by 8 area of memory 164 having a block set for each line color and blank for each background color space within the rectangle 163. A row 164 corresponds

to row 80 in Fig. 7, and the compression from the FF rrepresentation to a single bit is already shown in this illustration.

Using the compression scheme set forth here the changed data is then run-
5 olength encoded, packaged with appropriate protocol and sent to the remote device through the various layers as necessary.

In the preferred embodiment we also remove any screen pointers from a
screen to be sure that they do not appear as part of the signal data before doing the
10 screen capture operation and then with the pointer data there can be replaced, on top of the display of the screen data. Thus the video RAM has an image of the screen without any pointers drawn on it, and the X,Y screen coordinate location allows the operating program to draw the pointers/cursors on the screees where appropriate, painting over the background or any other feature at that X,Y coordinate.

15 It should be noted that other data compression methods, such as run length encoding may be superimposed upon any screen capture data to minimize the amount of data necessary for transfer for large screen areas that are totally one color, or an other, or are a simple stippled pattern.

20 Fig. 9b is a representation of data being communicated between the remote expert and patient sites, having, as shown here a train of data packets 101 - 103 each organized to have its own unique identifying header 104 - 106, respectively. Preferably it is organized as follows. ECG Data 95 is treated as its own data type and sent separately in its own packets. ECG Data 101 can be treated 102 (dotted
25 line) as Other Screen data 103. Other screen data 103 is raw screen pixel data with no special features identified, and in the preferred embodiment will contain only the changes made from the last time the screen was updated.. Pointer Location data 97 is communicated between expert and patient sites.

to row 80 in Fig. 7, and the compression from the FF rrepresentation to a single bit is already shown in this illustration.

Using the compression scheme set forth here the changed data is then run-
5 olength encoded, packaged with appropriate protocol and sent to the remote device through the various layers as necessary.

In the preferred embodiment we also remove any screen pointers from a
screen to be sure that they do not appear as part of the signal data before doing the
10 screen capture operation and then with the pointer data there can be replaced, on top of the display of the screen data. Thus the video RAM has an image of the screen without any pointers drawn on it, and the X,Y screen coordinate location allows the operating program to draw the pointers/cursors on the screes where appropriate, painting over the background or any other feature at that X,Y coordinate.

15 It should be noted that other data compression methods, such as run length encoding may be superimposed upon any screen capture data to minimize the amount of data necessary for transfer for large screen areas that are totally one color, or an other, or are a simple stippled pattern.

20 Fig. 9b is a representation of data being communicated between the remote expert and patient sites, having, as shown here a train of data packets 101 - 103 each organized to have its own unique identifying header 104 - 106, respectively. Preferably it is organized as follows. ECG Data 95 is treated as its own data type and sent separately in its own packets. ECG Data 101 can be treated 102 (dotted
25 line) as Other Screen data 103. Other screen data 103 is raw screen pixel data with no special features identified, and in the preferred embodiment will contain only the changes made from the last time the screen was updated.. Pointer Location data 97 is communicated between expert and patient sites.

Referring now to Fig. 11, a typical PD (Patient site located Device) display screen 110 is illustrated as it would appear on the remote expert's screen. As in most computing devices, there currently exists a background tool bar main menu T.B.m. For this invention we have determined that it is advantageous to use a number of active icons here labeled AI's 1-n. In this display the icon AI1 opens a phone directory when 'clicked' or activated by a cursor, that may allow interaction with the communications programs, or just monitoring of the connection with the expert's device. AI2 and AI3 provide indications of how often the updates occur between the remote and local screen displays, and can be used to change that frequency in preferred embodiments. The next two, AI4 and AI 5 are icons which will be active depending on whether the expert indication pointer is active (AI4 would be highlighted in that case) or whether the expert's enabling pointed would be active, here shown as an icon of a light wand. AI6 indicates the print function, and the designer can include whatever others are desired for a given system set up. The title bar TB, shows that the PD is engaged in a session, or prepared to run a session.

Note there is an arrow Ar, which the expert user has only on his screen, and does not appear on the screen of the patient located programmer. This arrow can be used by the expert to perform auxiliary tasks in background like receiving a data dump (as illustrated in Fig. 14) or for consulting an expert system resident on his computing device or other system on his computer, while at the same time the expert may continue to monitor the PD location.

Here we also have a number of other active buttons on the screen appropriately labeled with their functional title 112-118. For example, the implanted medical device in communication with the local PD here is indicated at site 111, as a Minix 8340 a type of pacemaker. The serial number or other unique identifier could of course be displayed also if desired, perhaps by clicking on the

button labeled Minix 8340. This screen 110 shows the main menu 112. Three sections 113a-c are present which can report data relative to those areas or enable options under button 113c, as well. Using button 116f, for example, the ECG screen area 160 would become active and show the real time ECG form the leads on the patient's body or from the telemetered out information from the implanted device or from a pacing system analyzer if such a device is active and attached to the programmer.

In Fig 12, the expert and the local user can be seen to both have their controlled icons, 119a and 119b, respectively, concentrating on a portion of the ECG display, the expert's pointing to the left of the patient local user's.

Note that the icons for each location are different and unique to that location, even though they may be displayed remotely. This reduces the potential for confusion and enhances accurate and fast communication of knowledge of the situation between the remote users.

On screen 130 in Fig. 13, the menu option displaying pacing parameters 122 is present. It was made active by the expert using his button 135 at the location shown. The expert would then have switched pointers and activated the active button press at the patient local site using indicator 134, indicating a rate of 49 bpm in area 131. Depending on the embodiment, the user may have to activate this rate by moving his own patient local button to this location shown by the expert at 134 or the expert could activate the local programmer if desired. The ability to allow a local expert to actually run the features of the programmer rather than asking the patient local person to do what the remote expert indicates is a matter of preference, and it may be considered better in some situations to keep the remote expert from having the power to modify local programming or it may be considered essential to disallow such distant control.

For the purpose of being complete, note Fig. 14 which has a separate window superposed over the background window of Figs. 11-13. This merely indicates that normal functionality of multithreaded or multiprocessing systems may be maintained in a background mode at either site. The large white space area displays the data dump that informs the expert about the system and the patient he is communicating about. This data dump screen need not be displayed at the patient site, as in the preferred embodiment. To keep this data dump display (or any other local display) completely local, we handle the covered area of the screen in a manner similar to the way we handled the screen area covered by a pointer. That is to maintain the buffer data for the underlying screen separately from the data for the local-only display in relation to a set of X,Y registers that locally maintain the reference to the appropriate screen location. The data in the local-only buffer is then "painted over" the appropriate screen area in accordance with the display routine for the local operating system.

Here the activating pointer or cursor 141 is an arrow, and if it is clicked when located over one of the buttons 144 or 144a, that function will be run by the computer supporting this display 143.

To reiterate and further explain how this pointer coordination is accomplished, we can use a set of registers with the X,Y coordinates of the pointers for both machines at both locations. When any data is received from a remote location, and there is any indication of a pointer move in that data, the registers will be compared with the incoming data for the relevant registers and if a change is found, the program will accommodate it in accord with the appropriateness of the action and by moving the cursor related to that changed data at the local screen. Flow charts in Figs. 10a and b clarifies this matter. In a preferred embodiment we used a standard O S/2 programmer window implementation, the drawing screen driven by the client window procedure and any re-drawing of all or parts of the window is done automatically when the client window procedure receives a

WM_PAINT message from the system. (OS/2 is an operating system for IBM corporation computers). This WM_PAINT message used by OS/2 indicates that all or part of the window is invalid and needs to be redrawn. In our implementation, the screen drawing is driven from the individual application and not from a central procedure. This makes it difficult to restore the screen image under the second pointer as it is moving across the screen. A user touch screen may also be implemented and the pointer may be moved by a remote pointer message received from the remote PC if a remote mouse move is received from the remote PC the programmer will change the icon's pointer image to the remote PC representation and then move the pointer to be coordinates defined in the mouse move message. Moving the pointer at the programmer will be ignored and a pointer message will be returned to the remote computer PC with the Coordinates of the screen position being touched at the programmer site.

Generally then, a single pointer can be viewed and used by both the expert location and the programmer. It should be noted that the pointer on the 9790 programmer is touch screen activated. The pointer/cursor is displayed as an icon. It is coordinated, transferred, and displayed separately from the screen image. The icon is removed from the programmer screen, the screen, with the icon removed is what is captured, and the pointer-icon data then is painted over this screen. Messages relating to the data about the pointer/cursor icons must be sent back and forth between the two remote systems when either a touch-screen event occurs on the programmer or the user selects and moves the pointer/cursor icon(s) at the remote expert device. This is done with an input hood which monitors system messages for touch screen or pointer event messages. If a pointer/cursor activation event occurs, the X,Y coordinate is compared to the previous event saved at both sites for that pointer. If the X,Y coordinates are different, the pointer-icon displayed is moved to the new location and a message is sent to the remote device which allows the remote device to do the same. The pointer-icon representing the programmer is then displayed on both systems telling both users that the

programmer was the last to move the pointer. This will display the effect of dragging a pointer icon at the remote location, when it is dragged at the local location. In one preferred embodiment, the programmer user's movement of a pointer icon takes precedence over that of the remote user and discards remote user commands if they are received while the programmer user is manipulating his local icons. This assures stable operation at the local site, allowing the physician at the programmer to maintain control to enable his rapid response to occurrences at the patient.

Referring now to Fig. 10a, the operation of either system to a remote command or screen change is outlined in flow chart 200. The incoming data is received 201, and separated into it's component parts of relevance in step 202. For example, the ECG portion of data in block 95 (From Fig. 9) is forwarded without it's header to step 203, and the process of updating the ECG or other physiologic signal waveform is accomplished. The pointer data from block 97 is sent to the process performed at step 204, and if no change is found 207, nothing further is done with that data. Likewise, the screen data areas are checked for changes (steps 205, 213 and updates of the relevant changes are made (step 213). If command information is received, it is transferred to processes at steps 209, 210 which determine if activation of these commands is appropriate, then the appropriate activation occurs, step 211, and then the display is again updated as needed 213.

In Fig 10b, the local control program steps that monitor and send out changes needed to be noticed by the remote system is outlined in flow chart 250. In the normal state, the operating system program is monitored such that the ECG buffer is maintained and locally updated 252. Any local commands initiated by pointer/cursor icon activation and movements of these pointers are kept track of 253, and any remote inputs related to any of these just mentioned data buffers may be received 258. A part of the inventive application program will then update all screen area indexes with the information from steps 252, 253, and 258, in step 254,

in accord with the detailed description above. A separate portion of this program will maintain and update the new pointer/cursor locations in step 255 so these can be painted over the display without affecting the underlying display screen buffer. For local only information regarding the screen display, the same manner of operation is maintained as for the pointer locations. A part of the monitoring program performs step 256 which checks to see whether there have been any changes locally that need to be sent to the remote system, or simply sends the relevant buffer information on the occurrence of a time-out, if preferred. The relevant data is then packaged and sent 257 and the system returns to the normal monitoring state.

While it will be apparent to the reader that many adaptations may be made of this disclosure, it is only limited by the following claims.

*******APPENDIX*******

/* These three code fragments have been excerpted from actual code files. They are presented here to illustrate a preferred embodiment

5 Screen Capture processing procedure for finding changed areas of screens and allowing for compression of video RAM data which can then be sent to a remote location without loss of waveform fidelity*/

/* Main screen processing loop */

10 void

 mnt_ScreenTransfer::threadFn()

 {

 NG_TRACE_FUNCTION();

15 unsigned int index;

 int minx, maxx, miny, maxy;

 int new_minx, new_maxx, new_maxy;

 int x, y;

20 unsigned int checked;

 bool doneCheckingScreenWorth;

 unsigned int lastLeftOffIndex = 0;

 prt_ScreenInfo* p_screenInfo;

 DATETIME dt;

25 mnt_Ulong byte_length;

 /***** debug collection *****/

 /* open a file for debug collection */

 if (v_debugOn) {

```
p_v_ScreenFile = fopen("SCREEN.DAT", "w");
fclose(p_v_ScreenFile);          /* discard contents */
p_v_ScreenFile = fopen("SCREEN.DAT", "w");

5      }

    /****** end debug collection *****/

    try
    {
10      // printf("The ScrnThread is executing.\n");
      p_screenInfo = new prt_ScreenInfo;

      v_screenThreadIsActive = TRUE;
      //tbd add to main the call to declare time-to-become-inactive

15      while(v_screenThreadIsActive)
      {
        doneCheckingScreenWorth = FALSE;
        DosSleep(500L);

20      // Check to see if there is a change in environment
      // For example, maybe color was turned off
      checkEnvironment();

25      screenCapture();
      screenCompare();

      /****** debug collection *****/
      if (v_debugOn) {
```

// %4d means 4 places of decimal, %2x means 2 places of hex, % means
an output

// Note to output that a screen capture/compare was done

fprintf(p_v_ScreenFile, "%.2d: %.2d: %.2d. %.2d screen
capture/compare\n",

dt.hours, dt.minutes, dt.seconds, dt.hundredths);

fflush(p_v_ScreenFile);

// In order that debug file does not grow indefinitely, close after
// two minutes of collection.

DosGetDateTime(&dt); // Get Date & Time

if ((dt.minutes - v_dateTime.minutes) > 2) {

fflush(p_v_ScreenFile);

fclose(p_v_ScreenFile);

v_debugOn = FALSE;

}

}

/*****DEBUG CODE COLLECT END *****/

// Find Rectangles and send

while (!doneCheckingScreenWorth)

{

index = lastLeftOffIndex;

y = index / v_bytesPerRow;

x = index - (y * v_bytesPerRow);

minx = -1;

for (checked = 0; checked < v_screenSizeInBytes; checked++)

```
{
    // Check flags until we have a rectangle to send.
    if (v_screenChangedFlagBuffer[index])
    {
5        // If we don't have a rectangle yet set one.
        if (minx == -1)
        {
            minx = maxx = x;
            miny = maxy = y;
10        }
        else
        {
            // Check the x and y gap spacing to see if we can include
            // this in the current rectangle.
15        if ((x >= minx - v_gapAllowedX) &&
            (x <= maxx + v_gapAllowedX) &&
            (y >= miny - v_gapAllowedY) &&
            (y <= maxy + v_gapAllowedY))
            {
20        // Determine the size of the new proposed rectangle.
            // We will not set this as the current rectangle
            // until we know that the area is not too large
            // for a single message.
            new_minx = minx;
25        new_maxx = maxx;
            new_maxy = maxy;
            if (x < new_minx)
            {
                new_minx = x;
30        }
            }
```

```

else if (x > new_maxx)
{
    new_maxx = x;
}

```

5

```

// Since we are advancing in the y direction and we don't
// allow wrapped rectangles. We only need to adjust maxy.
if (y > new_maxy)
{
    new_maxy = y;
}

```

10

```

// Is the proposed block too big for a single message?
// Note that following doesn't work if right side set to less

```

15

than max

```

if (((new_maxy - miny + 1) * (new_maxx - new_minx +
1)) > v_maxSingleMessageSize)

```

```

{
    /***** debug collection *****/

```

20

```

    if (v_debugOn) {

```

```

        /* %4d means 4 places of decimal, %2x means 2
places of hex, % means an output */

```

```

        fprintf(p_v_ScreenFile, "        block-too-big test,
%6d <= index %5dx %5dy\n",

```

25

```

        index, x, y);

```

```

        fflush(p_v_ScreenFile);

```

```

    }

```

```

    /*****DEBUG CODE COLLECT END

```

```

*****/

```

30


```

        // Send the current block, and set an index
        // to remember where we left off.
        lastLeftOffIndex = index;
        break;
5      }
    else
    {
        // Make the current block equal to the proposed block.
        minx = new_minx;
10      maxx = new_maxx;
        maxy = new_maxy;
    }
}
else
15 {
    // If we went too far in the y direction,
    // send the current block, and look for the next
    // block where we left off.
    if (y > maxy + v_gapAllowedY)
20 {
        /****** debug collection *****/
        if (v_debugOn) {
            /* %4d means 4 places of decimal, %2x means 2 places of
hex, % means an output */
25      fprintf(p_v_ScreenFile, "          gap test\n");
            fflush(p_v_ScreenFile);
        }
        /******DEBUG CODE COLLECT END
******/
30

```

```

        lastLeftOffIndex = index;
        break;
    }
}
5      }
    }

    // Advance to the next coordinate.
    x++;
10    if (x >= v_bytesPerRow)
    {
        x = 0;
        y++;
        if (y >= v_maxScreenPixelsY)
15        {
            y = 0;
        }
    }

20    // Need to wrap back to the beginning?
    index++;
    if (index >= v_screenSizeInBytes)
    {
        /***** debug collection *****/
25        if (v_debugOn) {
            /* %4d means 4 places of decimal, %2x means 2 places of hex,
            % means an output */
            fprintf(p_v_ScreenFile, "        doneScreenWorth/wrap
test\n");
30            fflush(p_v_ScreenFile);

```

```
    }

    /*******DEBUG CODE COLLECT END
******/
5      // Added
      doneCheckingScreenWorth = TRUE;
      index = 0;

      // If we found some changes, send that area. Otherwise,
10     // setting index=0 leads to looking for more changes
      //at the top of the screen.
      if (minx != -1)
      {
          lastLeftOffIndex = index;
15         break;
      }
  }

  } // end "for" checking all up to v_screenSizeInBytes

20

  /******* "break" statements on "for" above end up here
******/

      // Did we find a rectangle?...indicated by minx not being -1
      if (minx != -1)
25     {
          byte_length = getScreenRect(minx * v_pixelsPerByte,
                                      miny,
                                      (maxx + 1)*v_pixelsPerByte-1,
                                      maxy);
30
```

```

p_screenInfo->header.x1 = minx * v_pixelsPerByte;
p_screenInfo->header.y1 = miny;
p_screenInfo->header.x2 = (maxx + 1)*v_pixelsPerByte-1;
p_screenInfo->header.y2 = maxy;
5      p_screenInfo->header.skip = 0;// number of horizontal lines that
are skipped between each line

p_screenInfo->header.format = v_colorFormat;
p_screenInfo->header.compression = prc_comp_rlCompression;
// Set pointer to rectangle data.
10      p_screenInfo->p_data = v_sendRectangleBuffer;

/***** debug collection *****/
if (v_debugOn) {
    /* %4d means 4 places of decimal, %2x means 2 places of hex, %
15      means an output */
    DosGetDateTime(&dt);          // Get Date & Time
    fprintf(p_v_ScreenFile, "%.2d: %.2d: %.2d. %.2d ",
    dt.hours, dt.minutes, dt.seconds, dt.hundredths);
    fprintf(p_v_ScreenFile, "sending
20      Index => %7dlenght => %5d %6d %6d %6d %6d \n",
        lastLeftOffIndex, byte_length, minx, maxx, miny, maxy);
    fflush(p_v_ScreenFile);
}
/*****DEBUG CODE COLLECT END *****/

25      // Send the current screen rectangle.
// For now ignoring the return from sendScreenData call
v_p_screenDataMsg->sendScreenData(p_screenInfo,
v_pixelsPerByte);

```

```
        } // end if we found a rectangle (i.e. minx != -1)
    } // end while notDoneCheckingScreen's Worth
} // end while screenThreadActive
```

5

```
delete p_screenInfo;
```

```
// printf("The ScrnThread is exiting.\n");
```

```
}
```

10

```
catch (...)
```

```
{
```

```
    // No Return
```

```
    SYD_systemError(IString("mnt_Screen in catch...some uncaught  
exception"));
```

15

```
}
```

```
}
```

```
/* Function to read video memory */
```

20

```
void
```

```
mnt_ScreenTransfer::screenCapture()
```

```
{
```

```
    NG_TRACE_FUNC();
```

25

```
    unsigned char    *p_dive = NULL;
```

```
    unsigned char    *p_current = NULL;
```

```
    unsigned int      line = 0;
```

```
    unsigned int      i = 0;
```

```
    unsigned char      mono_pixels = 0;
```

30

```
// remove the pointer from the screen.
```

```
WinShowPointer(HWND_DESKTOP, false);
```

```
5      p_dive= (unsigned char *)v_p_diveFrameBuffer;  
      p_current= v_currentScreenBuffer;
```

```
      if ((v_pixelsPerByte == 8) && (v_platform ==  
syt_Programmer::c_486Color9790)) {
```

```
10      // BW on Color 9790 Screen capture  
      for(line = 0; line < v_maxScreenPixelsY; line++)  
      {  
          for(i = 0; i < v_maxScreenPixelsX; i++)  
15      {  
          mono_pixels < <= 1;  
          if (*p_dive++)  
          {  
              mono_pixels |= 1;  
20      }  
  
          if ((i % 8) == 7)  
          {  
              *p_current++ = mono_pixels;  
25      }  
          }  
          p_dive+= mnc_486CUnusedDiveSection;;  
      }  
      } // end on if 486C and color off
```

```
else if (v_pixelsPerByte == 8) {

    // 386/486EL platform 9790 Screen capture
    for(line = 0; line < v_maxScreenPixelsY; line++)
5      {
        for(i = 0; i < v_bytesPerRow; i++)
        {
            *p_current++ = *p_dive++;
        }
10     }
} //end if 386/486EL

else if (v_pixelsPerByte == 1)
{
15

    // Color Screen capture
    /* Copy data from the frame buffer to the memory bitmap. */
    for(line = 0; line < v_maxScreenPixelsY; line++)
    {
20     for(i = 0; i < v_maxScreenPixelsX; i++)
        {
            *p_current++ = *p_dive++;
        } // end "for" to copy x

25     p_dive+ = mnc_486CUnusedDiveSection;;
    } // end "for" to copy y

    } // end if 486C and color on

30     else {
```

```
SYD_systemError(IString("in screen compare and v_pixelsPerByte not 8  
nor 1") +
```

```
    IString(v_pixelsPerByte));
```

```
    }
```

5

```
    // display the pointer on the screen.
```

```
    WinShowPointer(HWND_DESKTOP, true);
```

```
}
```

10

```
/* Function to find changed rectangles */
```

```
void
```

```
mnt_ScreenTransfer::screenCompare()
```

15

```
{
```

```
    NG_TRACE_FUNC();
```

```
    unsigned int index;
```

```
    unsigned char current;
```

20

```
    // printf("The screenCompare is executing.\n");
```

```
    index = 0;
```

```
    while (index < v_screenSizeInBytes)
```

25

```
    {
```

```
        // compare algorithm is to compare and if different,
```

```
        // copy "current" into "previous" and set a
```

```
        // flag in screenChangedFlagBuffer
```

```
        current = v_currentScreenBuffer[index];
```

30

```
        if (current != v_previousScreenBuffer[index])
```



```
        {  
            v_previousScreenBuffer[index] = current;  
            v_screenChangedFlagB  
uffer[index] = 0xFF;  
5         }  
            index ++;  
  
        } // end while index < screenSizeInBytes.  
  
10     // printf("The screenCompare is exiting.\n");  
    }
```

What is claimed is:

1. A programmer device for use in a medical communications system for communicating in near real time information from between at least two remote sites, one remote site at a patient location having a programmer device which has a programmer generated display, and an other (of said at least two remote sites) at an expert location having a computing device with a computing device generated display, each location having available substantially similar screen displays, such that information related to an implantable medical device and a patient at said patient location remote site can be reviewed simultaneously on said substantially similar displays at said at least two remote sites facilitated by data communications transferred across a communication line between said at least two remote sites, said programmer device being configurable to perform various operations by executing programs through a microprocessor and memory system and comprising:

a physiologic signal device connection, connected to said programmer device for receiving electrical output representing a patient's physiologic signal waveform ,and for producing a signal suitable for use by a programmer screen display operating program for generating a display from said electrical output at at least one display area including at least a physiologic waveform signal display area and producing instantaneous changes in said at least one display area, at least one of said changes representative of changes in said waveform display,

said programmer display operating program, employing at least;

a first video display data buffer for holding a video data representation of information displayed in one of said at least one display areas of said programmer generated display,

a second video display buffer for holding data representing instantaneous changes in at least one of said areas,

a display generating program for using a data image of said video data representation in said first video display buffer for producing said programmer generated display,

5 an index generating program for reviewing data in said first video display buffer and creating a first index buffer corresponding to said at least one of said display areas, said index generating program indicating in said first index buffer by storing data therein representing which of all locations in said at least one area have changed from a previous review, and for capturing a data representation of said
10 changed locations in a capture buffer and,

a sending program for packaging data comprising contents of said first index buffer and said capture buffer in appropriately encoded format, and for sending said appropriately formatted index buffer data across said communication line.

15
2. A programmer device as set forth in claim 1, said display generating program further comprising a two color display generating program for interpreting said physiologic signal waveform as two color display data, and wherein said index
20 generating program has a data compression program for receiving said physiologic waveform index data and coding it such that for each byte, one bit is set to indicate whether a change has occurred in that byte and such that for the entire physiologic waveform index buffer, a compressed physiologic waveform index is prepared.

25
3. A system as set forth in claim 1 wherein said programmer operating program further comprises;

a pointer/cursor coordination program having a buffer for holding at least
30 location information for each pointer/cursor to be coordinated, and having a

monitoring means for updating any programmer generated display change in pointer/cursor activation and location, said monitoring means updating each said pointer/cursor buffer with any change in location information from a last review, said location information being changeable responsive to pointer/cursor user controls based on a local user changing location status of any pointer/cursor under local control, and wherein said pointer/cursor coordinating program further comprises a buffer for containing information regarding at least location status of each remote pointer that is to be displayed on said patient location display, and

wherein said patient location screen display operating program further comprises;

a program for receiving new information for each said remote pointer/cursor buffer and operating on such data to produce a display of said remote pointer/cursor in accord with the data in said remote pointer/cursor buffer such that any changes in the location data of a remote pointer/cursor buffer results in a change in location on the display of that remote pointer/cursor on the patient location display screen.

4. A programmer as set forth in claim 3 wherein said pointer cursor operating program further comprises an activation buffer for maintaining activation status for each pointer/cursor capable of activation.

5. A programmer device as set forth in claim 2, wherein said sending program further comprises a pointer data capture program for capturing and then for sending the data in said pointer/cursor buffer to said communication line.

6. A system as set forth in claim 3 further comprising;

a program for monitoring which of all said pointer/cursors is active such that said active cursor/pointer may initiate a command, and for maintaining a data buffer containing such monitored information.

7. A computing device operating program for coordinating a physiologic waveform display with a programmer generated display as set forth in claims 1 - 6,
5 comprising

communication means to receive data sent by said programmer device and to produce a physiologic waveform display therefrom.

10 8 A computing device operating program for displaying an image substantially the same as the display on the programmer generated display having communications means to receive data sent by said programmer device set forth in claims 1-6 and generating a substantially similar display therefrom.

15 9. A system for displaying substantially the same image on at least two remote locations having a programmer device and a computing device as set forth in claim

10Method for communicating between at least two remote locations so as to enable coordinated display of a physiologic waveform and pointer/cursor information
20 comprising;

capturing pointer/cursor information regarding a location on a display screen for each said pointer/cursor,

redacting image data from a video display buffer based on said
pointer/cursor information so that the remaining video buffer data represents a
25 display sans said pointer/cursors

reviewing the redacted video image buffer so as to locate any changes since a last review,

creating an index of information describing which locations in said video buffer contain changed data since said last review,

capturing representations of the images at the changed locations and storing this as data in a buffer,

compressing the index and the capture buffer data,

5 sending the compressed index and capture buffer data from one of said at least two locations.

11. A method as set forth in claim 10 wherein said pointer/cursor data is sent in said sending step.

10 12. A method of communicating between at least two remote locations so as to enable coordinated display of a physiologic waveform and pointer/cursor information comprising:

receiving compressed index and capture data across a communication line from one of said at least two locations at a first of said at least two locations,

15 decompressing said index and capture buffer data,

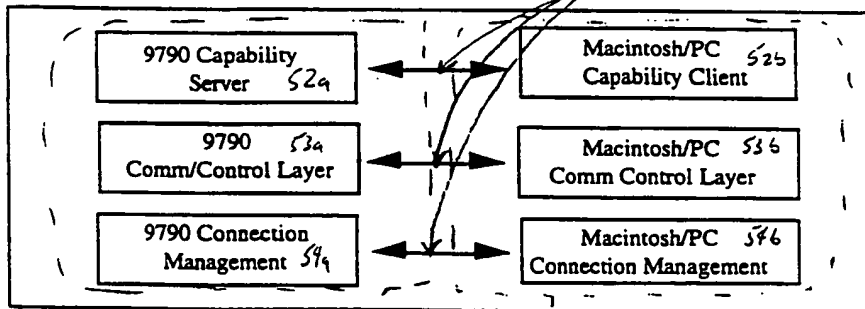
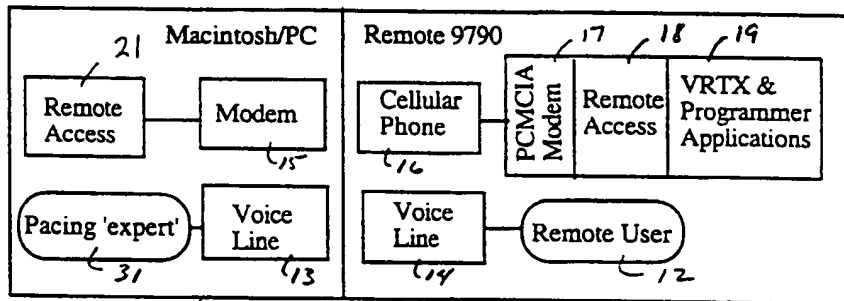
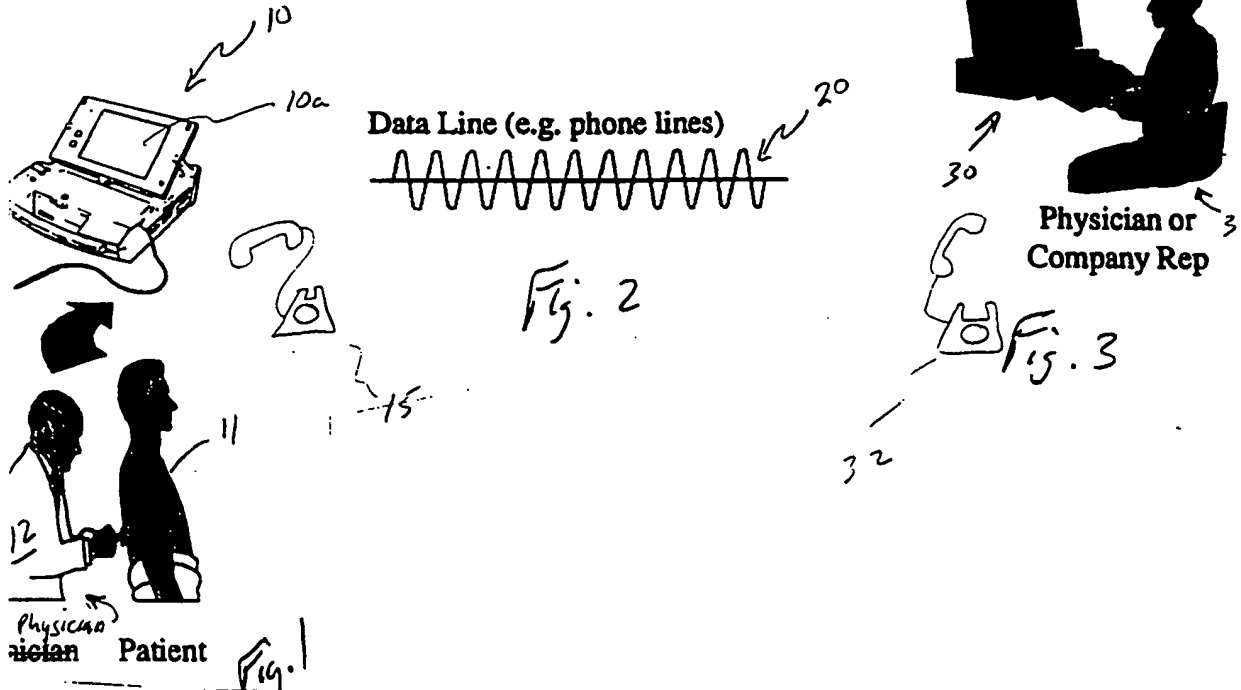
swapping in information from said capture buffer data at locations in a video display buffer representing corresponding locations to those indicated in said index data so as to enable a display at said first location to display said changes.

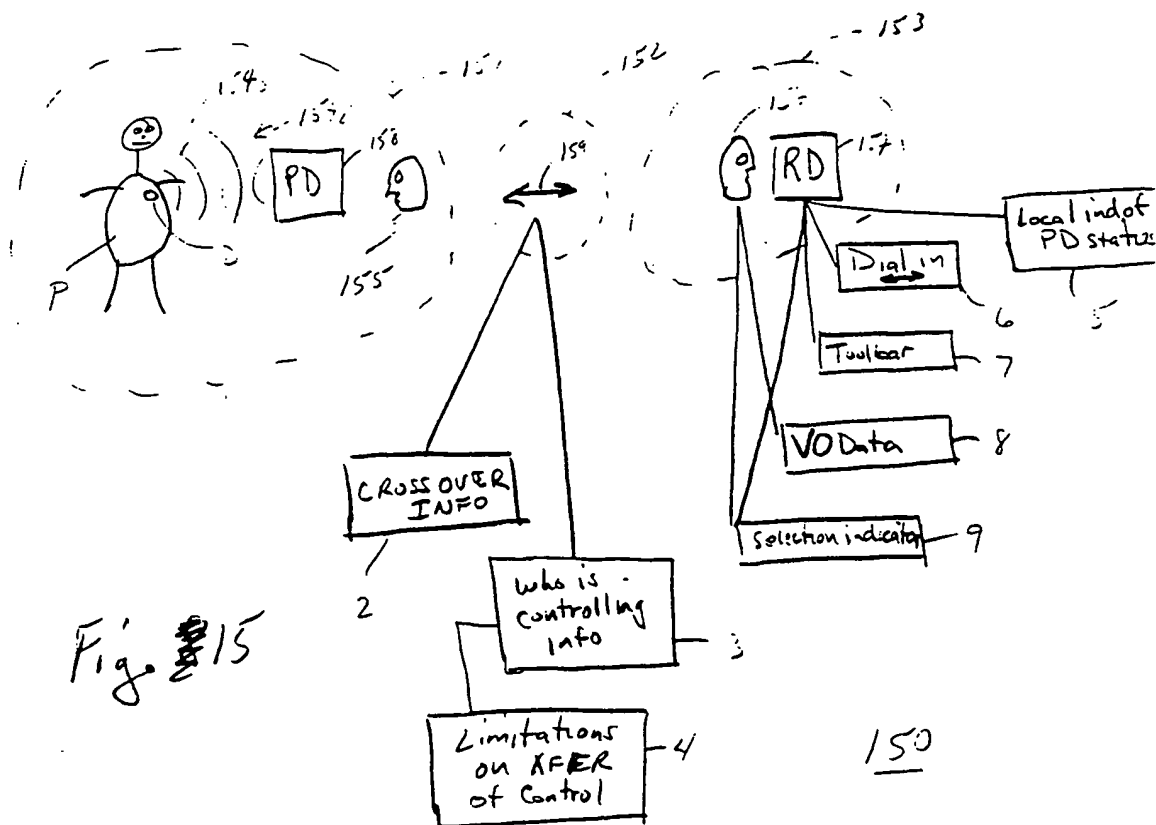
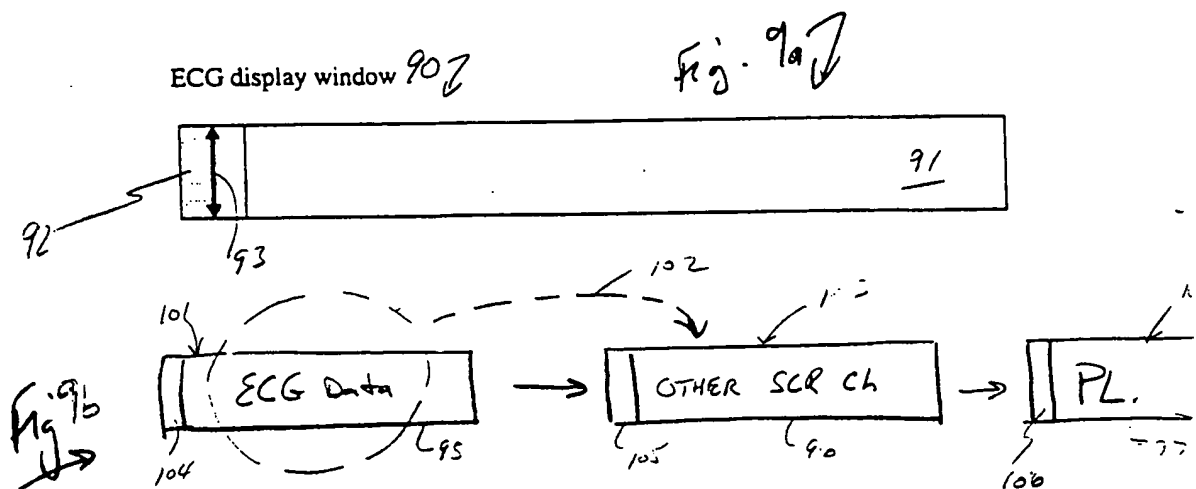
20 13. A method as set forth in claim 12 further comprising the step of receiving data representing pointer/cursor location on a display screen from said one location of at least two remote sites and still further comprising the step of displaying an iconic image of a pointer/cursor representing a pointer displayed at said one remote location in a location on the screen in said first locations indicated by the received
25 representative pointer /cursor location data.

1/11

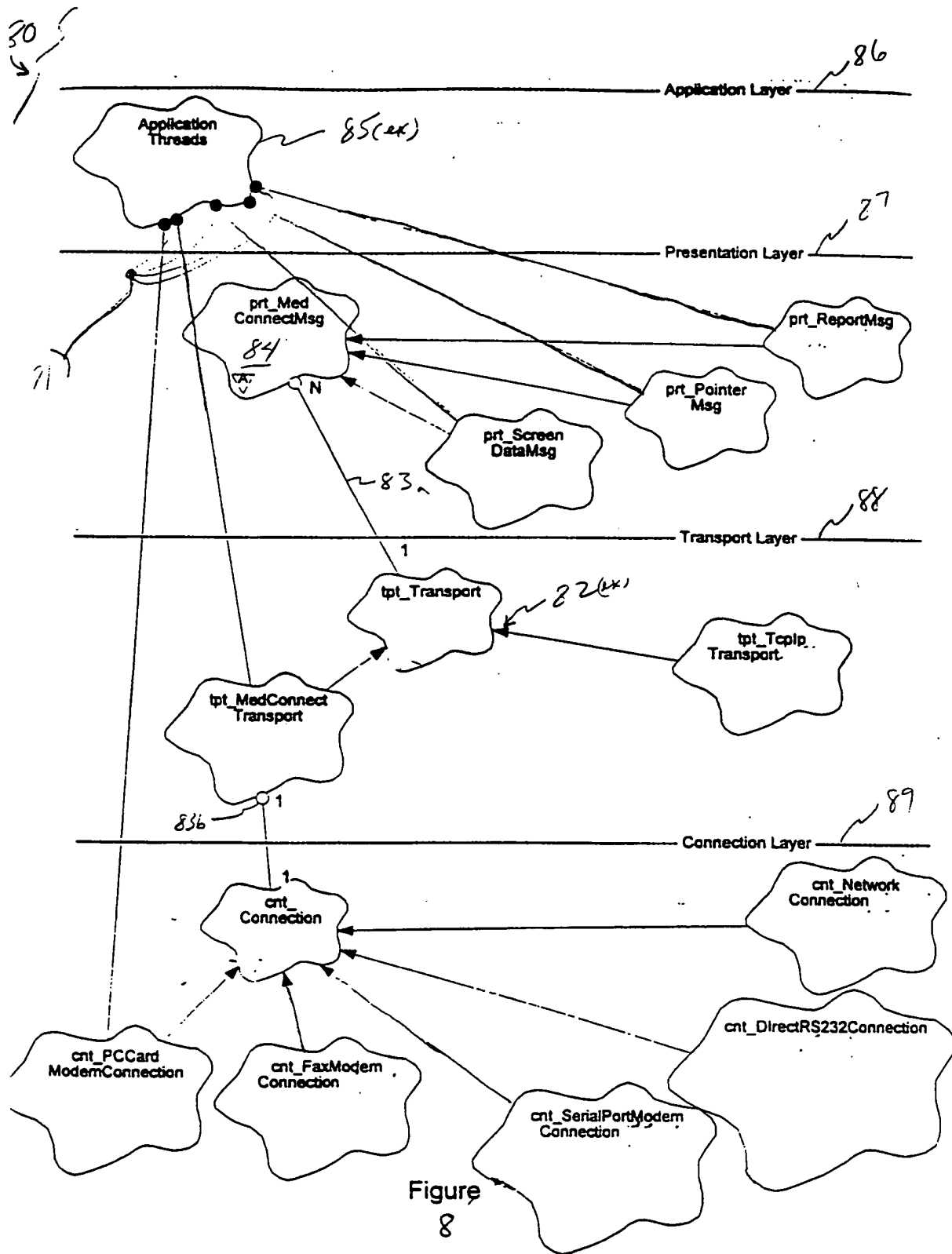
Programmer

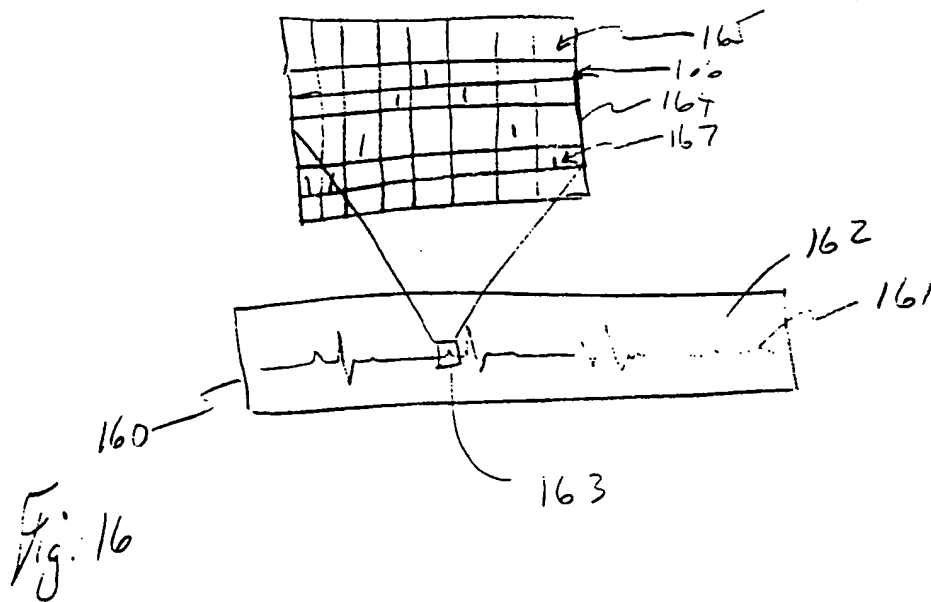
Remote "Experts"



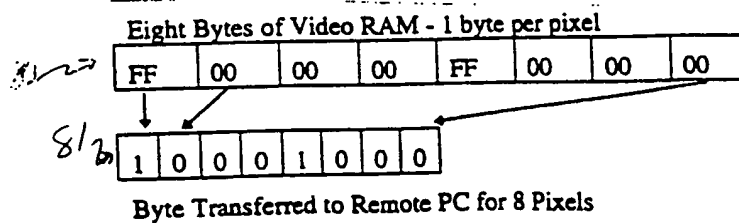
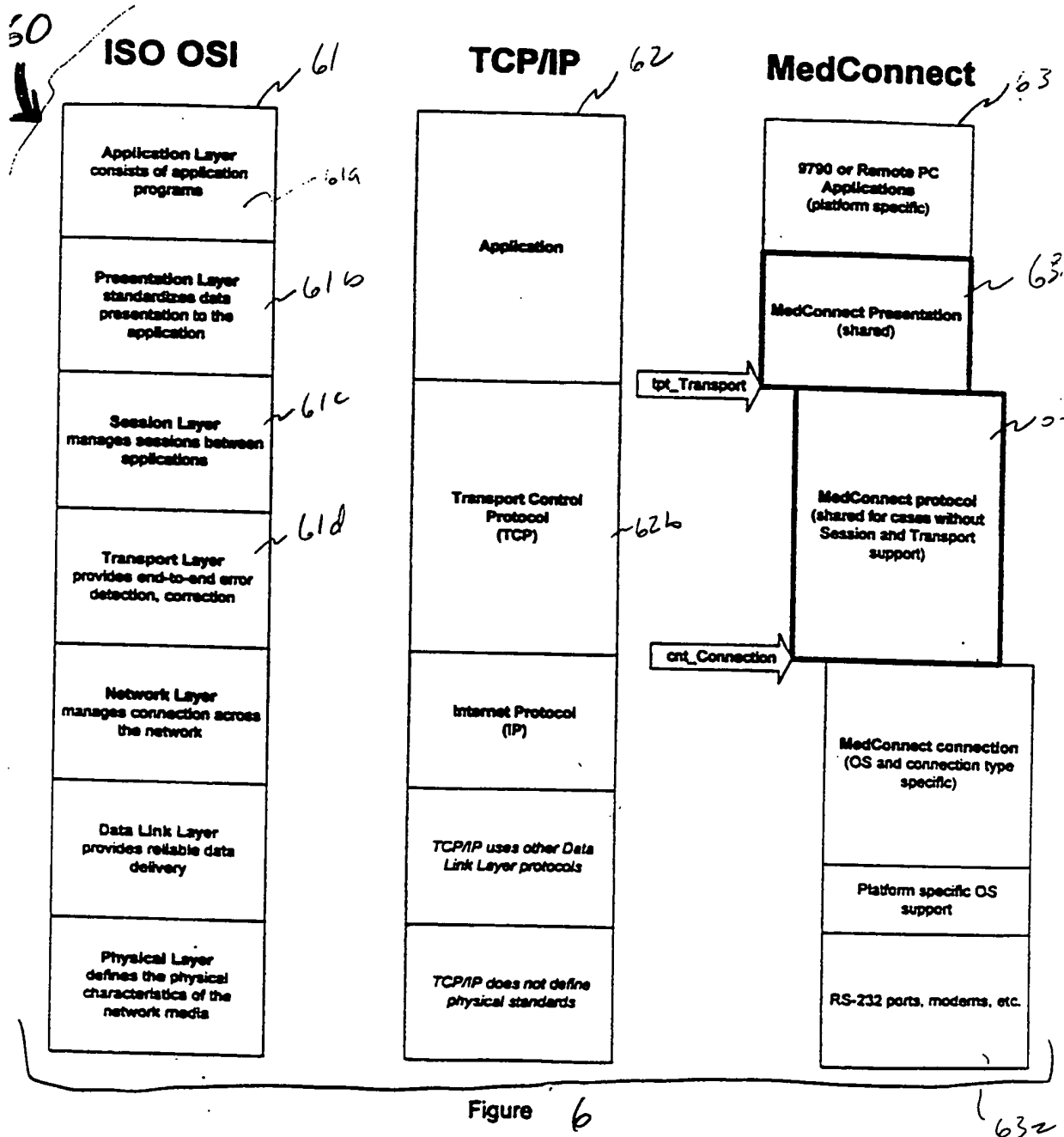


3/11





5/11



6/11

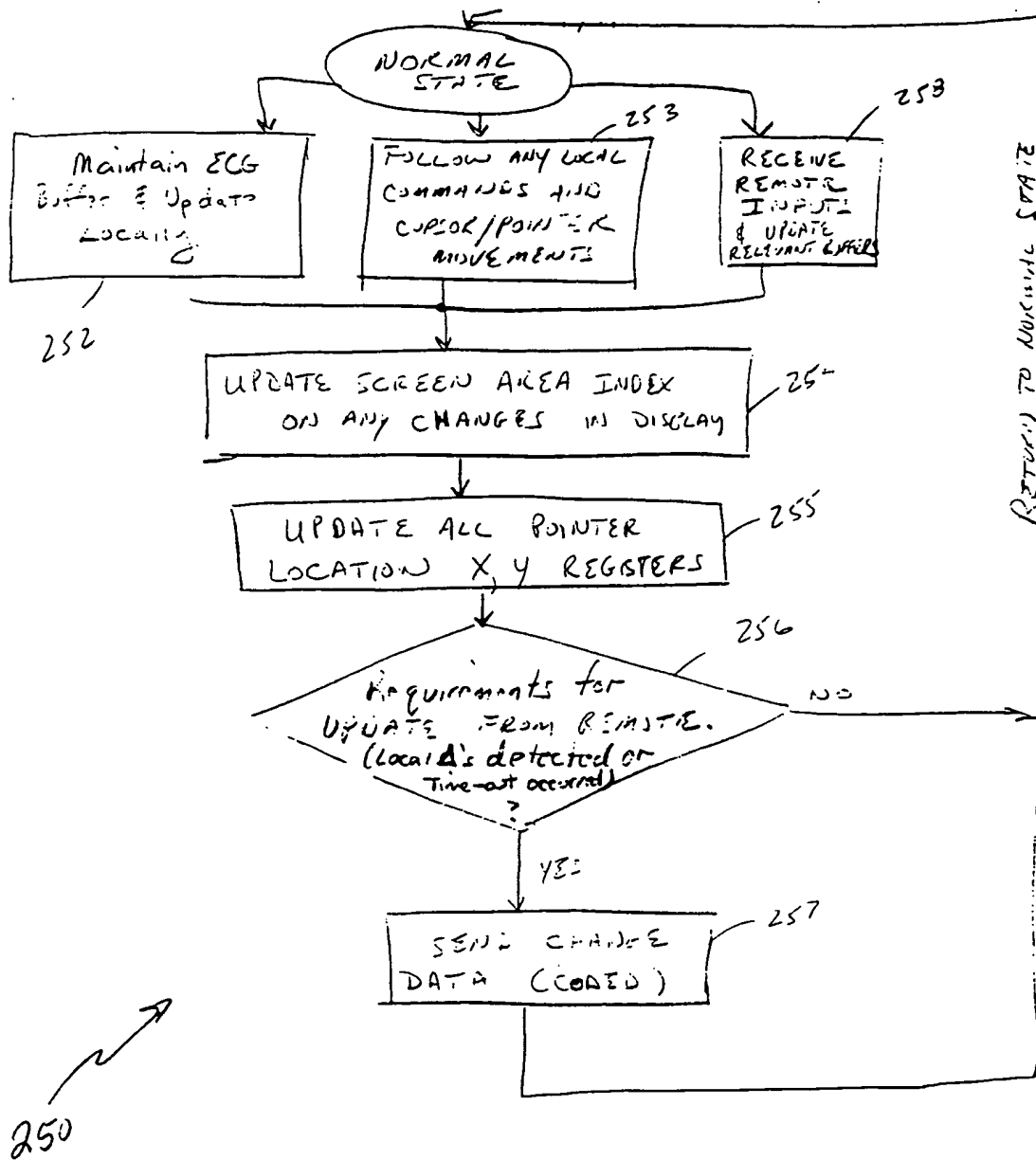


Fig 10b

LOCAL CONTROL PROGRAM

7/11

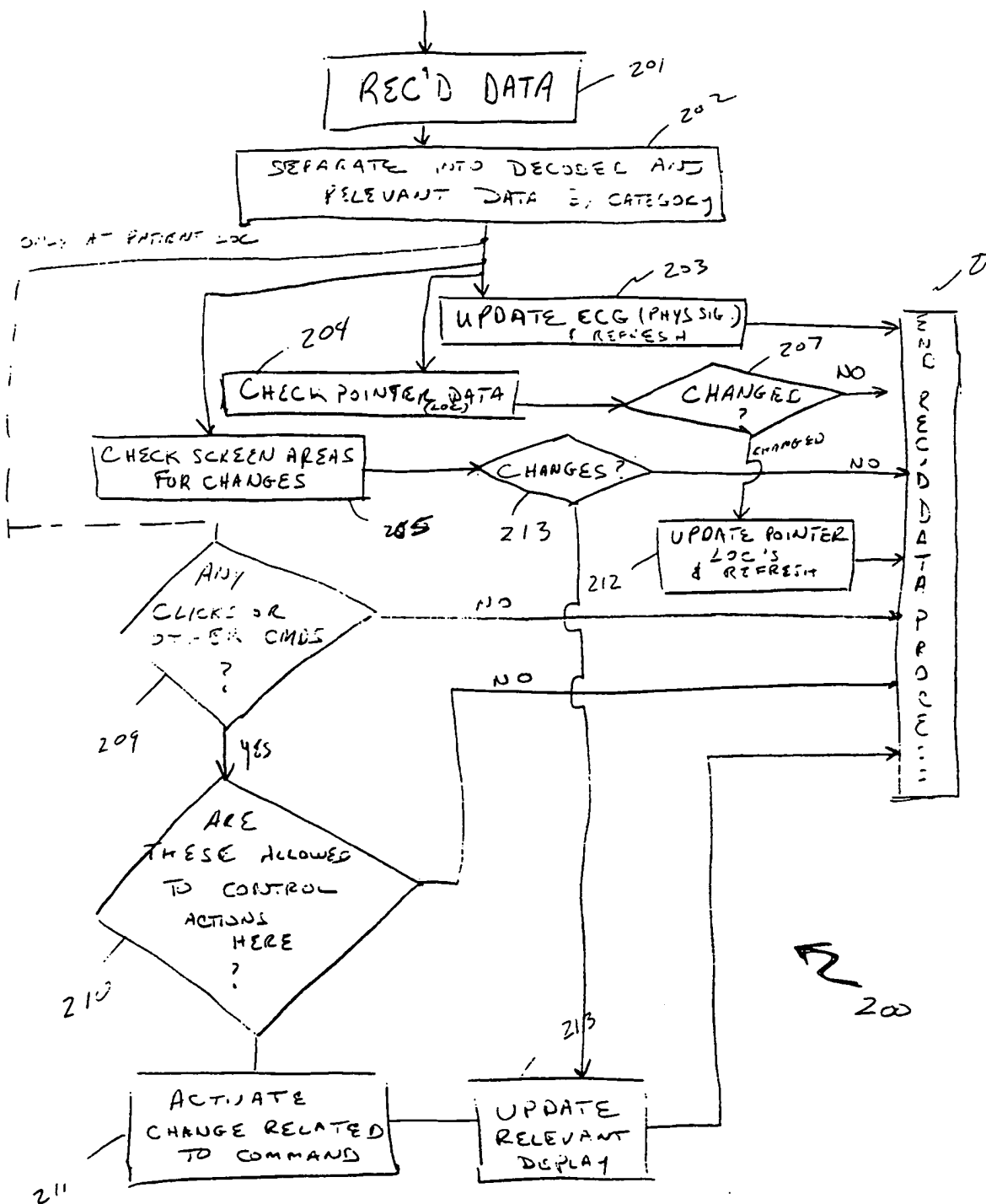
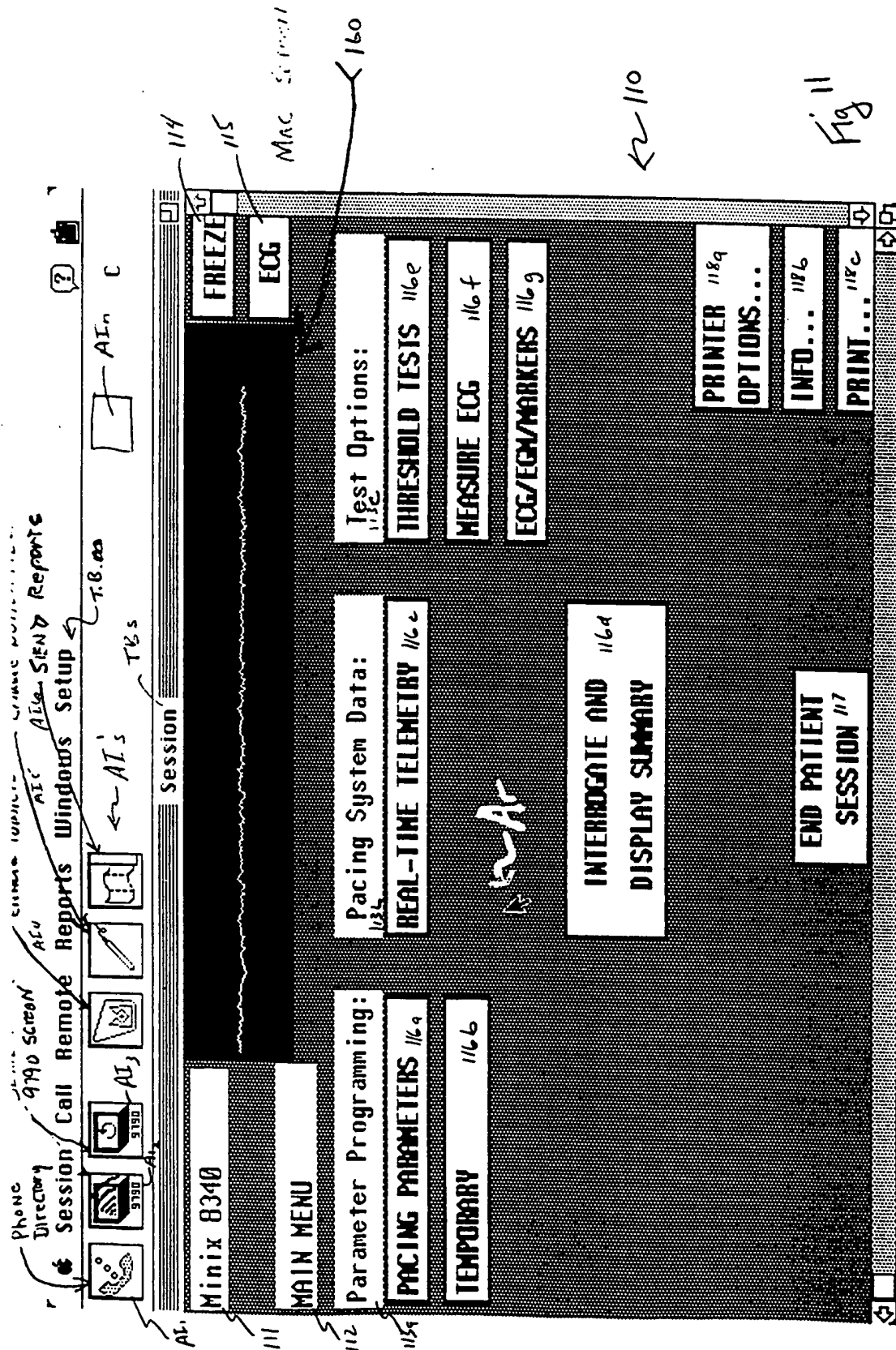


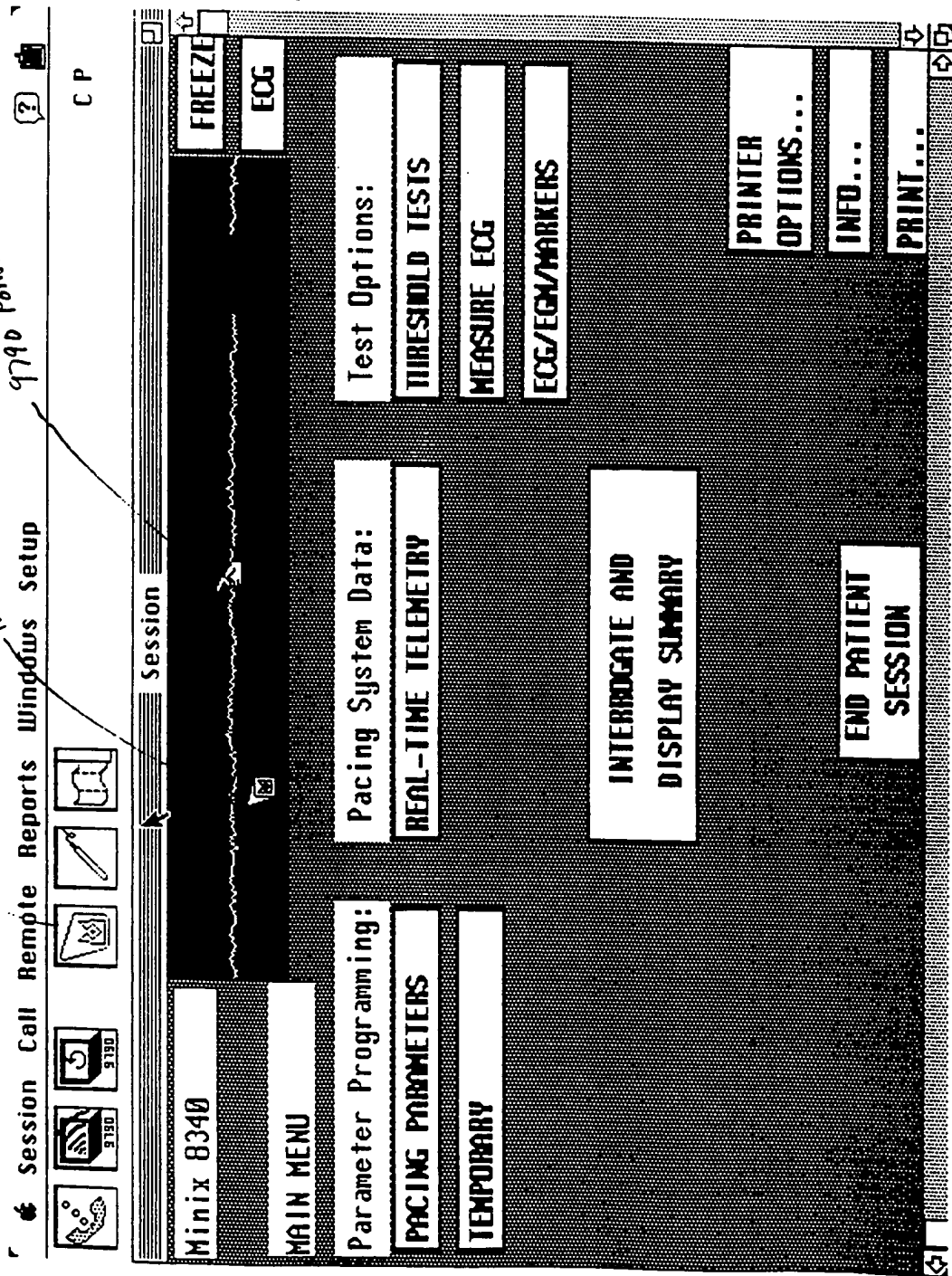
Fig 10a REMOTE RESPONSE.

8/11



9/11

MAC "pointer" 119a
9790 "pointer" 119b



10/11

Session Call Remote Reports Windows Setup

CP

Minix 8340

PACING PARAMETERS

Pacing Mode ¹³⁵

Rate ¹³² 70 ppm

Hysteresis OFF bpm

Refractory 325 ms

Amplitude 5.0 V

Pulse Width 1.5 ms

Sensitivity 2.5 mV

Polarity Bipolar

Status Reset ¹³⁶

PENDING ¹²³

UI 70

OFF

325

5.0

1.5

2.5

Bipolar

Rate

44 51 58 65 72 79

45 52 59 66 73 80

46 53 60 67 74 81

47 54 61 68 75 82

48 55 62 69 76 83

49 56 63 70 77 84

50 57 64 71 78 85

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

137

136

135

134

133

132

131

130

129

128

127

126

125

124

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

84

83

82

81

80

79

78

77

76

75

74

73

72

71

70

69

68

67

66

65

64

63

62

61

60

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

40

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Session

FREEZE

EOG

MAC SCREEN

"Button Press" Selected

134 indicates Button Press is Active.

133

130

132

131

135

136

137

123

122

121

120

119

118

117

116

115

114

113

112

111

110

109

108

107

106

105

104

103

102

101

100

99

98

97

96

95

94

93

92

91

90

89

88

87

86

85

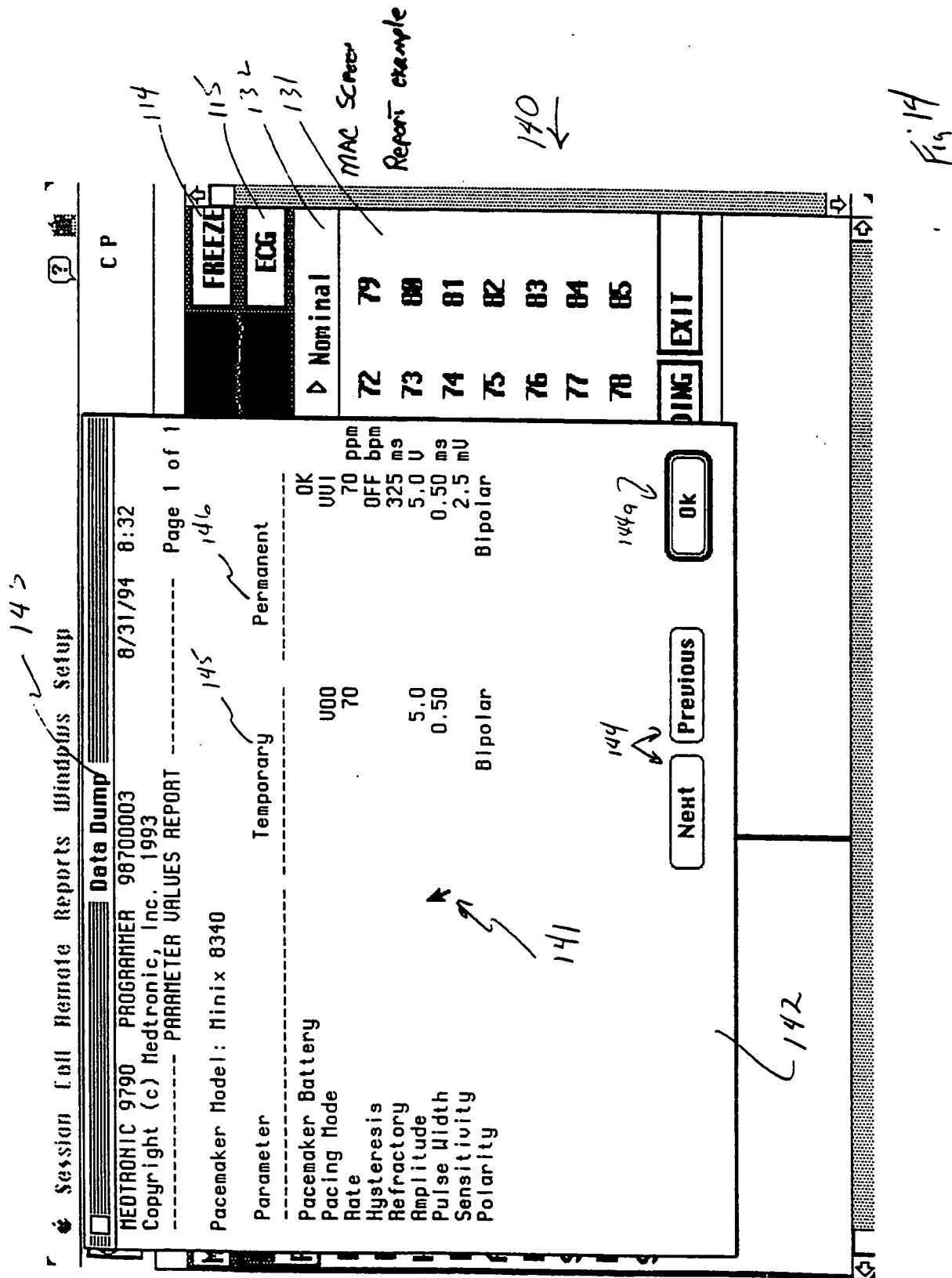
84

83

82

81

11/11



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US98/06085

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : A61N 1/37

US CL : 128/903, 905; 607/ 26, 27, 32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 128/903,905; 607/26,27,32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, E	US 5,752,977 A (GREVIOUS et al.) 19 May 1998, cols. 1-5, col. 7 lines 16-42, and col. 11 line 47 to col. 19.	1-13
Y	US 5,487,755 A (SNELL et al.) 30 January 1996, Abstract, cols. 1-8, col. 14 lines 14-46, and cols. 17-25.	1-13
Y	US 5,487,754 A (SNELL et al.) 30 January 1996, entire document.	1-13
Y	US 5,431,691 A (SNELL et al.) 11 July 1995, entire document.	1-13
Y	US 5,309,919 A (SNELL et al.) 10 May 1994, entire document.	1-13



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

16 JUNE 1998

Date of mailing of the international search report

20 JUL 1998

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

STEPHEN HUANG

Telephone No. (703) 308-3399